

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 2002216121 A

(43) Date of publication of application: 02.08.02

(51) Int. Cl

G06T 1/20

G06F 15/16

(21) Application number: 2001012553

(22) Date of filing: 22.01.01

(71) Applicant:

MATSUSHITA ELECTRIC IND CO LTD

(72) Inventor:

NAKAO SHINYA
UEDA HIDEJI
ITSUZAKI YOSHIHIRO
GOKAN MANABU
TANAKA SHOICHI

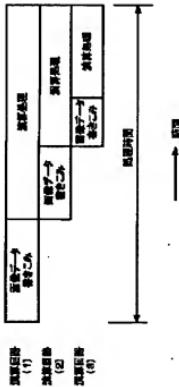
(54) OPTIMIZING METHOD OF PARALLEL IMAGE PROCESSING

(57) Abstract:

PROBLEM TO BE SOLVED: To provide an optimizing method of parallel image processing capable of enhancing a degree of parallelism in a parallel image processor to perform arithmetic processings of image data to be transferred in parallel.

SOLUTION: The degree of parallelism of the arithmetic processings is enhanced by allocating the number of image data to be stored in memories of the respective arithmetic operation circuits so that the arithmetic processings by the respective arithmetic circuits are simultaneously completed.

COPYRIGHT: (C)2002,JPO



(19)日本国特許庁 (JP)

(12)公開特許公報 (A)

(11)特許出願公開番号

特開2002-216121

(P2002-216121A)

(43)公開日 平成14年8月2日(2002.8.2)

(51)Int.Cl.*

G 0 6 T 1/20
G 0 6 F 15/16

識別記号

6 1 0

F I

G 0 6 T 1/20
G 0 6 F 15/16

テ-マコト*(参考)

C 5 B 0 4 5
6 1 0 F 5 B 0 5 7

(21)出願番号

特願2001-12553(P2001-12553)

(22)出願日

平成13年1月22日(2001.1.22)

(71)出願人

000005821
松下電器産業株式会社
大阪府門真市大字門真1006番地

(72)発明者

中尾 真也
大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72)発明者

植田 秀司
大阪府門真市大字門真1006番地 松下電器
産業株式会社内

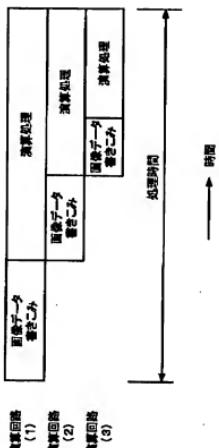
(74)代理人

100068087
弁理士 森本 翠弘

最終頁に続く

(54)【発明の名称】 並列画像処理最適化方法

(57)【要約】

【課題】 転送される画像データを並列に演算処理する
並列画像処理装置において、演算処理の並列度を高める
ことができる並列画像処理最適化方法を提供する。【解決手段】 各演算回路による演算処理が同時に完了
するよう各演算回路のメモリに蓄積する画像データ数を
割り当てることによって、演算処理の並列度を高める。

【特許請求の範囲】

【請求項1】プロセッサとプロセッサからアクセスされるメモリとを備えた演算回路を複数個具備し、順次転送される画像データを前記各演算回路のメモリに蓄積し、前記各演算回路のプロセッサがメモリに蓄積された画像データに対する演算処理を行う並列画像処理装置において、前記各演算回路による画像データの演算処理が同時に完了するよう前記各演算回路のメモリに蓄積する画像データ数を割り当てる特徴とする並列画像処理最適化方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、FA等の画像処理応用分野において、CCDカメラ等の画像入力手段を用いて撮像される画像データを複数のプロセッサを用いて並列に演算処理する並列画像処理装置における並列画像処理最適化方法に関する。

【0002】

【従来の技術】従来の並列画像処理方法について図面を用い、以下に説明する。図3は画像入力手段を含めた従来の並列画像処理装置の構成の一例を示す図である。また、図4は、図3に示す演算回路4の内部構成図であり、図5は画像入力手段であるCCDカメラ1により撮像される画像の走査線をイメージ化した図である。

【0003】CCDカメラ1は、対象物を撮像すると図5の矢印で示す走査線11の順番に従ってアナログの映像信号をADコンバータ2に出力する。ADコンバータ2に入力されたアナログの映像信号は標準化され、時系列に変化する8ビットのデジタルデータに変換される。以下、この映像信号をデジタルデータに変換したものを画像データと記述する。

【0004】アナログの映像信号は、ここでは図5の走査線方向(水平方向)に512分割される。以下、この512個の画像データを水平ラインと記述する。1画面はこの水平ラインを垂直方向に480本並べる事により構成される。従って、画面上の座標は水平方向の座標x、垂直方向の座標yにより表すことができる。この様*

$$P(y) = \sum_{x=0}^{511} d(x, y)$$

投影データ $P(y)$ は水平ラインの数だけ計算される。

【0012】演算完了後、投影データ $P(y)$ はプロセッサ8によりデュアルポートメモリ10に書き込まれる。デュアルポートメモリ10はプロセッサ8と図3のCPU5の両方からアクセスが可能なメモリである。

【0013】CPU5は各演算回路4内のデュアルポートメモリ10よりこの投影データを読み出し、合成して1画面の投影データを得る。以下、各プロセッサ8における演算処理時間を考える。

【0014】各演算回路4のプロセッサ8は同一の性能を有するものとする。プロセッサ8の内部はバイオペ

*子を図5に併せて示す。

【0005】画像データはバスコントローラ(マスター)3に転送される。バスコントローラ(マスター)3は画像バスを介して各演算回路4に対して画像データを順次転送する。

【0006】従来の並列画像処理方法においては、各演算回路4が均等な配分の演算処理を行うよう、1画面分の画像データを3等分して160本の水平ラインずつ各演算回路4のデータメモリ7(図4参照)に書き込むようにする。つまり、図6に示すように、演算回路(1)が0~159水平ラインの画像データを、演算回路(2)が160~319水平ラインの画像データを、演算回路(3)が320~479水平ラインの画像データを演算処理するようする。

【0007】以下、各演算回路4の処理動作について図4を用いて説明する。バスコントローラ(マスター)3からの画像データの転送先である各演算回路4のバスコントローラ(スレーブ)6は、画像バスを転送される画像データのデータ数をカウントすることによって所定の範囲の画像データをデータメモリ7に書き込む。このときバスコントローラ(スレーブ)6は、プロセッサ8からのデータメモリ7に対するアクセスを禁止し、画像バスからの画像データの書き込みを許可している。

【0008】画像データの書き込みが完了した時点でバスコントローラ(スレーブ)6は画像バスからのアクセスを禁止し、プロセッサ8に対して画像データの書き込みが完了したことを通知するとともに、プロセッサ8のデータメモリ7へのアクセスを許可する。

【0009】画像データの書き込み完了を受けてプロセッサ8はデータメモリ7に書き込まれた画像データに対する演算処理を開始する。該並列画像処理装置における演算処理は水平方向への投影データ計算とする。水平方向への投影データ計算は以下の演算を行う。

【0010】座標(x, y)における画像データをd(x, y)、投影データをP(y)とすると、

【0011】

【数1】

(数1)

40 構造となっており、プログラム実行とプログラムフェッチを同時に実行できる。また、プログラムメモリ9はデータメモリ7と独立にアクセスされ、更にプログラムメモリ9に記憶されているプログラムデータはプロセッサ内蔵の高速なキャッシュメモリに格納される。このためプログラムフェッチ時間はプログラム実行時間に影響を及ぼさず、プログラム実行時間がプロセッサ8における演算処理時間となる。

【0015】プロセッサ8におけるプログラム実行時間は各命令(データメモリ7へのアクセス等)の実行時間を足し合せた値であり、該並列画像処理装置における各

命令の実行時間は、データメモリ7、デュアルポートメモリ10へのアクセスを2クロック、プロセッサ8内部の演算を1クロックとする。また、投影データP(y)、垂直カウンタy、水平カウンタx、データメモリ7から読み出された後の画像データd(x, y)はプロセッサ8内部のレジスタに格納されているものとする。

【0016】プロセッサ8の行う演算処理のフローチャートを図7に示す。各ステップの実行クロックサイクル数は以下の通りである。

ステップS7001 垂直カウンタ(y)を0に初期化…1クロック
 ステップS7002 投影データP(y)を0に初期化…1クロック
 ステップS7003 水平カウンタ(x)を0に初期化…1クロック
 ステップS7004 データメモリより画像データd(x, y)リード…2クロック
 ステップS7005 P(y) + d(x, y)を計算し、計算結果をP(y)とする…1クロック
 ステップS7006 水平カウンタ(x)インクリメン特開2002-216121

$$40 \times 5 = 200 \text{ (ナノ秒)}$$

となる。よって1水平ライン512個の画像データの演算処理に要する時間は
 $200 \times 512 = 102400 \text{ (ナノ秒)}$

である。

【0018】ステップS7002、ステップS7003、ステップS7008、ステップS7009、ステップS710※
 $40 \times 6 = 240 \text{ (ナノ秒)}$

を要する。

【0019】ステップS7001の演算処理は160水平ライン分の演算処理に対し、1クロックサイクルつまり40ナノ秒を要する。これと(数3)、(数4)より、1★

$$(102400 + 240) \times 160 + 40 \\ = 16422440 \text{ (ナノ秒)}$$

となる。

【0020】統いて、該並列画像処理装置が1画面について画像処理するためには要する時間は計算する。例えば画像データの画像転送クロックを10MHzとする。1クロックで1画像データがデータメモリ7に書き込まれ☆

$$100 \times 512 \times 160 = 8192000 \text{ (ナノ秒)}$$

を要する。

【0021】各演算回路4の画像データ書き込みと演算処理のタイミングを図8に示す。図8からわかるとおり、演算回路(1)、(2)は演算処理完了後アイドル状態となり、最後に処理を開始する演算回路(3)の処理が完了するまで全体の処理は完了しない。◆

$$8192000 \times 3 + 16422440 \\ = 40998440 \text{ (ナノ秒)} \approx 4.1 \text{ (ミリ秒)}$$

となる。

【0023】このように、従来の並列画像処理装置にお50の時間が演算回路間で異なるため、最後に処理を開始す

*ト…1クロック

ステップS7007 水平カウンタ値を判定し、未終了(x < 511)ならステップS7004へ分岐…1クロック

ステップS7008 P(y)をデュアルポートメモリに格納…2クロック

ステップS7009 垂直カウンタ(y)インクリメント…1クロック

ステップS710 垂直カウンタ値を判定し、未終了(y < 159)ならステップS7002へ分岐…1クロック

該並列画像処理装置が1画面について画像処理する場合において、1個のプロセッサ8が要するプログラム実行時間(演算処理時間)を計算する。

【0017】ステップS7004～ステップS7007の合計5クロックサイクルの演算処理は1画像データに対して実行される。プロセッサ8のクロック周波数を25MHzとすると1クロック当たり40ナノ秒であるので、1画像データに対する5クロックサイクルの演算処理時間は

間は

(数2)

となる。よって1水平ライン512個の画像データの演算処理に要する時間は

(数3)

※は1水平ラインの演算処理につき1回実行される。合計6クロックサイクルであるからその演算処理には

(数4)

★60水平ラインの演算処理、即ち該並列画像処理装置が1画面について画像処理する場合において、1個のプロセッサ8が要するプログラム実行時間は

☆るとすると、1画像データ書き込みに要する時間は100ナノ秒である。従って、1個のプロセッサ8の処理する512×160個の画像データをデータメモリ7に書き込むには、

(数5)

◆【0022】該並列画像処理装置において、演算回路(3)の処理が完了する時間は、全演算回路が画像データをデータメモリ7に書き込む時間と、演算回路(3)による160水平ラインの演算処理時間とを足し合わせたものであり、

(数7)

いって、データメモリへの画像データ書き込み完了までの時間が演算回路間で異なるため、最後に処理を開始す

る演算回路の処理が完了するまで全体の処理は完了しない。そのため並列画像処理装置における演算処理の並列度が低下し、その結果、画像処理時間が長くなるという問題点がある。

【0024】

【発明が解決しようとする課題】本発明は、上記問題点に鑑み、各演算回路による演算処理が同時に完了するよう各演算回路のメモリに蓄積する画像データ数を割り当てるによって、演算処理の並列度を高めることができる並列画像処理最適化方法の提供を目的とする。

【0025】

【課題を解決するための手段】本発明における請求項1記載の並列画像処理最適化方法は、プロセッサとプロセッサからアクセスされるメモリとを備えた演算回路を複数個備えし、順次転送される画像データを前記各演算回路のメモリに蓄積し、前記各演算回路のプロセッサがメモリに蓄積された画像データに対する演算処理を行う並列画像処理装置において、各演算回路による画像データの演算処理が同時に完了するよう前記各演算回路のメモリに蓄積する画像データ数を割り当てる特徴とする。

【0026】本発明によれば、演算回路を複数個備えし*

$$(102400 + 240) \times L + 40 \\ = 102640 \times L + 40 \quad (\text{ナノ秒}) \quad (\text{数8})$$

で計算される。

【0030】画像バスからの画像書き込み時間に対する演算処理時間の比をaとする。1個の画像データ書き込み※

$$100 \times 512 \times L = 51200 \times L \quad (\text{ナノ秒}) \quad (\text{数9})$$

となる。(数8) (数9)より、
 $a = (102640 \times L + 40) / (51200 \times L)$
 ≈ 2.0

が得られる。

【0031】演算回路(k)の処理する水平ライン数をLkライン(但し、k=1, 2, 3)とする。このとき、各演算回路4のデータメモリ7に対する画像データ★

演算回路(1)の演算処理完了時間は
 $51200 \times L_1 + a \times 51200 \times L_1 \quad (\text{数11})$

演算回路(2)の演算処理完了時間は
 $51200 \times (L_1 + L_2) + a \times 51200 \times L_2 \quad (\text{数12})$

演算回路(3)の演算処理完了時間は
 $51200 \times (L_1 + L_2 + L_3) + a \times 51200 \times L_3 \quad (\text{数13})$

演算回路(1)、(2)の演算処理完了時間が等しいす☆つ。

ると、(数11)、(数12)より以下の式が成立 ☆【0033】

$$51200 \times L_1 + a \times 51200 \times L_1 \\ = 51200 \times (L_1 + L_2) + a \times 51200 \times L_2 \quad (\text{数14})$$

これを計算すると、
 $L_2 = \{a / (1 + a)\} \times L_1 \quad (\text{数15})$

同様に演算回路(2)、(3)の演算処理完了時間が等◆り立つ。

しいすると、(数12)、(数13)より以下の式が成◆【0034】

$$51200 \times (L_1 + L_2) + a \times 51200 \times L_2$$

*た並列画像処理装置における演算処理の並列度を高めることができるとともに、装置コストを低減するために回路規模を小さくすることなどにも有用である。

【0027】

【発明の実施の形態】以下、本発明の実施の形態について図面を用いて説明する。本実施の形態においては、従来例と同様、図3、図4に示す構成をした並列画像処理装置を用いるものとする。また、プロセッサ8のクロック周波数や命令実行時間も従来例と同様である。画像転送クロックについても同様に10MHzで、1クロックサイクルで1画像データをデータメモリ7に書き込む。

【0028】本実施の形態においては、演算回路4間の演算処理並列度を高めるために、3個のプロセッサの演算処理が同時に完了するようデータメモリ7に蓄積する(書き込む)画像データ数、つまり水平ライン数を割り当てる。図1にこのときの各演算回路4の画像データ書き込みと演算処理のタイミングを図示する。以下にその水平ライン数割り当てる手順を示す。

【0029】従来例で示したように、演算装置における演算処理時間は演算装置が処理する画像データの範囲にほぼ比例する。例えば、水平ライン数をLとした場合、その演算処理時間は

※みに要する時間は100ナノ秒、1ラインの画像データ数が512個であるから、Lラインの画像データ書き込みに要する時間は、

$$100 \times 512 \times L \quad (\text{ナノ秒}) \quad (\text{数9})$$

★書き込み時間は51200×Lナノ秒である。

【0032】以上の条件で画像転送開始時から各プロセッサの演算処理完了までの時間を求める。

$$= 51200 \times (L1 + L2 + L3) + a \times 51200 \times L3 \quad (\text{数 } 16)$$

これを計算すると、

$$L3 = \{a / (1+a)\} \times L2 = \{a^2 / (1+a)^2\} \times L1 \quad (\text{数 } 17)$$

となる。このように演算回路 (1)、(2)、(3) の * の処理する画像データの水平ライン数の和は以下の通り処理する画像データの水平ライン数は等比数列となる。となる。

【0035】(数15)、(数17)より各演算回路 4 *

$$L1 + L2 + L3 = \{1 + a / (1+a) + a^2 / (1+a)^2\} \times L1 \quad (\text{数 } 18)$$

ここで(数10)より $a = 2.0$ であるから、

$$L1 + L2 + L3 = (19/9) \times L1 \quad (\text{数 } 19)$$

また、1画面480本の水平ラインを分割することから、

$$L1 + L2 + L3 = 480 \quad (\text{数 } 20)$$

(数19)、(数20)より、 $L1, L2, L3$ は以下 * 【0036】

の通り計算される。

*

$$L1 = 227.36 \quad (\text{数 } 21)$$

$$L2 = 151.57 \quad (\text{数 } 22)$$

$$L3 = 101.05 \quad (\text{数 } 23)$$

L は整数であるから小数点第一位を四捨五入することにより

$$L1 = 227 \quad (\text{数 } 24)$$

$$L2 = 152 \quad (\text{数 } 25)$$

$$L3 = 101 \quad (\text{数 } 26)$$

を得る。つまり、図2に示すように、演算回路 (1) は ★路4の総演算処理時間(画像データの書き込みと演算処理に要する時間)は以下の通りとなる。
 0~226水平ライン、演算回路 (2) は 227~37
 8水平ライン、演算回路 (3) は 379~479水平ラインの画像データを演算処理する。このときの各演算回路★

演算回路 (1)

$$51200 \times L1 + 102640 \times L1 + 40 = 34921720 \quad (\text{ナノ秒}) \quad (\text{数 } 27)$$

演算回路 (2)

$$51200 \times (L1 + L2) + 102640 \times L2 + 40 = 35006120 \quad (\text{ナノ秒}) \quad (\text{数 } 28)$$

演算回路 (3)

$$51200 \times (L1 + L2 + L3) + 102640 \times L3 + 40 = 34942680 \quad (\text{ナノ秒}) \approx 35 \quad (\text{ミリ秒}) \quad (\text{数 } 29)$$

となる。

【0038】このように、本実施の形態では画像処理時間が約3.5ミリ秒となり、各プロセッサに均等に画像データを割り当てる場合の画像処理時間である約4.1ミリ秒に比べて約14.6%の時間短縮が可能となる。☆

★【0039】一般に n 個の演算回路を用いて並列画像処理を行う場合を想定する。1水平ラインの画像データ書き込みに要する時間を T 、 k を 2 以上、 n 以下の整数とする

演算回路 $(k-1)$ の演算処理完了時間は

$$T \times \{L1 + L2 + \dots + L(k-1)\} + a \times T \times L(k-1) \quad (\text{数 } 30)$$

演算回路 (k) の演算処理完了時間は

$$T \times \{L1 + L2 + \dots + L(k-1) + Lk\} + a \times T \times Lk \quad (\text{数 } 31)$$

◆算処理完了時間が等しいという条件では、(数30)

となる。

【0040】演算回路 $(k-1)$ と演算回路 (k) の演◆ (数31) より以下の式が成立つ。

$$T \times \{L1 + L2 + \dots + L(k-1)\} + a \times T \times L(k-1)$$

$$= T \times \{L1 + L2 + \dots + L(k-1) + Lk\} + a \times T \times Lk \quad (\text{数 } 32)$$

(数32) より

$$Lk = \{a / (1+a)\} \times L(k-1) \quad (\text{数 } 33)$$

以上のように、本発明において n 個の演算回路を用いて 50 並列画像処理を行う場合には、各演算回路の処理する画

像データのライン数を等比数列の関係にすることによつて演算回路間の演算処理の並列度を高めることができる。

【0041】

【発明の効果】以上のように本発明によれば、演算回路間の演算処理の並列度を高めることができ、その結果、画像処理時間を短縮できる。これは画像処理完了までの時間に制限がある場合や、装置コストを低減するために回路規模を小さくする必要がある場合などに有用である。

【図面の簡単な説明】

【図1】本発明の実施の形態における各演算回路の画像データ書き込みと演算処理のタイミングを示す図

【図2】本発明の実施の形態における各演算回路の処理する画像データ数の説明図

【図3】画像入力手段を含めた従来の並列画像処理装置の構成の一例を示す図

【図4】従来の並列画像処理装置内の演算回路の内部構成図

【図5】CCDカメラにより撮像される画像の走査線を*20

*イメージ化した図

【図6】従来の並列画像処理装置内の各演算回路の画像データ書き込みと演算処理のタイミングを示す図

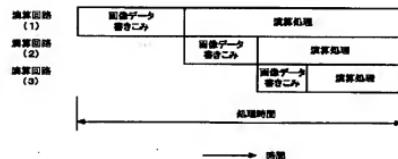
【図7】従来の並列画像処理装置内の各演算回路のプロセッサの行う演算処理のフローチャート図

【図8】従来の並列画像処理装置内の各演算回路の処理する画像データ数の説明図

【符号の説明】

1	CCDカメラ
2	ADコンバータ
3	バスコントローラ（マスター）
4	演算回路
5	CPU
6	バスコントローラ（スレーブ）
7	データメモリ
8	プロセッサ
9	プログラムメモリ
10	デュアルポートメモリ
11	走査線

【図1】



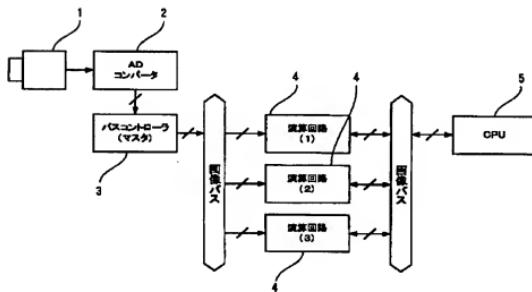
【図2】



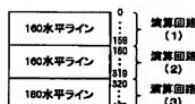
【図5】



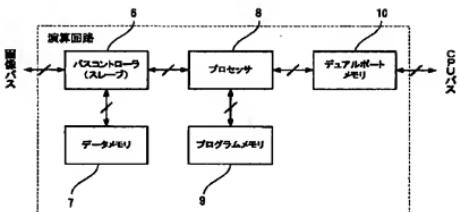
【図3】



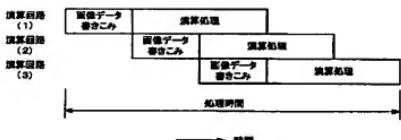
【図6】



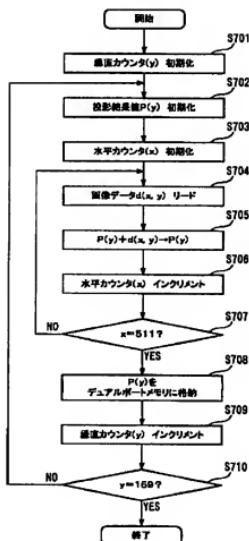
[図4]



[图 8]



[图7]



フロントページの続き

(72) 発明者 逸崎 嘉浩
大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72) 発明者 五関 学
大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72)発明者 田中 彰一
大阪府門真市大字門真1006番地 松下電器
産業株式会社内
Fターム(参考) 5B045 AA00 AA01 GG02 GG17
5B057 CA12 CA16 CB12 CB16 CH04
CH05 CH11 CH14